

Interrogation de la base

- Sélection
- Recherche
- Ordonner les réponses

- Fonctions de groupe
- Sous-requêtes
- Opérateurs de l'algèbre relationnelle

Sous-requêtes

- Sélection
- Recherche
- Ordonner les réponses

- Fonctions de groupe
- Sous-requêtes
- Opérateurs de l'algèbre relationnelle

Sous requêtes: Motivation

NUMAV	AVNOM	NBHWOL	CAPAV	LOC	COMP
200	Boeing	30	100	Marseille	AF
300	Heinkel	250	2	Allemagne	SING
500	Sportavia	300	1	CHINE	SING
600	Jet	2	300	Marseille	NY

Pour trouver les avions qui ont une capacité supérieure à celle du Boeing, une solution consiste à **récupérer la capacité du Boeing** et à la comparer avec la capacité des autres avions.

- Sélection
- Recherche
- Ordonner les réponses

- Fonctions de groupe
- Sous-requêtes
- Opérateurs de l'algèbre relationnelle

Sous requêtes

- Une sous-requête (ou requête imbriquée) est une requête incluse dans une autre requête (principale), i.e., la requête principale utilise le résultat de la sous-requête
- Les sous requêtes sont souvent utilisées dans les **clauses WHERE et HAVING**.

```
SELECT      exp, . . . , exp
FROM        table
WHERE       condition
(   SELECT  exp, . . . , exp
      FROM  table
      WHERE condition...) ;
```

- Sélection
- Recherche
- Ordonner les réponses

- Fonctions de groupe
- Sous-requêtes
- Opérateurs de l'algèbre relationnelle

Sous requêtes

- Sous requêtes Retournant au plus ligne
- Sous requêtes Multi-lignes
- Sous requêtes d'existence

- Sélection
- Recherche
- Ordonner les réponses

- Fonctions de groupe
- Sous-requêtes
- Opérateurs de l'algèbre relationnelle

Sous-requêtes: donnant au plus une ligne

Syntaxe:

```
SELECT ...  
FROM ...  
WHERE exp op  
(  SELECT exp  
    FROM ...  
    .... ) ;
```

Avec $op \in \{=, <>, <, <=, >, >=\}$

- Il y a une erreur d'exécution lorsque la sous requête ramène plusieurs lignes.

- Sélection
- Recherche
- Ordonner les réponses

- Fonctions de groupe
- Sous-requêtes
- Opérateurs de l'algèbre relationnelle

Sous-requêtes: donnant au plus une ligne

Exemples :

1. Pour connaître les vols qui utilisent le même avion que celui utilisé par le vol 101.

```
SELECT      NumVol
FROM        Vol
WHERE       NumAv = (    SELECT      NumAv
                          FROM        Vol
                          WHERE       NumVol = 101 ) ;
```

NUMVOL	NUMP	NUMAV
1	300	200
2	4001	400
490	7006	400
500	7006	300
101	300	400

2. Pour connaître les Pilotes qui sont dans la même ville que le pilote 300

```
SELECT  numP
FROM    pilote
WHERE   villeP = (    SELECT  villeP
                        FROM    pilote
                        where  numP=300) ;
```

NUMP	NOM	NBHVOL	COMP	VILLEP
1200	Gratien Viel	450 AF	Paris	
300	Didier Donsez	0 AF	Marseille	
4001	Richard Grin	1000 SING	Monaco	
7006	Fresnais	2450 SING	Nice	
200	martin	400 AC	(null)	
500	JACQUES	200 SING	Marseille	

- Sélection
 - Recherche
 - Ordonner les réponses

- Fonctions de groupe
 - Sous-requêtes
 - Opérateurs de l'algèbre relationnelle

Sous-requête donnant au plus une ligne

Syntax:

```
SELECT    ...
FROM      ...
WHERE    (exp1, ..., expN) op
(    SELECT exp1, ..., expN
        FROM ...
            ...
        ) ;
```

Avec op $\in \{=, <>\}$

Exemple :

Pour connaître les vols qui assurent le même trajet que celui du vol V101

- Sélection
- Recherche
- Ordonner les réponses

- Fonctions de groupe
- Sous-requêtes
- Opérateurs de l'algèbre relationnelle

Sous requêtes

■ Sous requêtes Retournant au plus ligne

■ Sous requêtes Multi-lignes

■ Sous requêtes d'existence

- Sélection
- Recherche
- Ordonner les réponses

- Fonctions de groupe
- Sous-requêtes
- Opérateurs de l'algèbre relationnelle

Sous requête donnant 0,1 ou plusieurs lignes

- Retourne plusieurs lignes
- Une requête imbriquée peut retourner une colonne entière
- Nécessite des opérateurs (par exemple pour comparer une valeur avec toutes celles retournées par la sous-requête)
 - IN , NOT IN
 - ANY, ALL

- Sélection
- Recherche
- Ordonner les réponses

- Fonctions de groupe
- Sous-requêtes
- Opérateurs de l'algèbre relationnelle

Sous requête donnant 0,1 ou plusieurs lignes

```
SELECT ....  
FROM ....  
WHERE exp [NOT] IN ( SELECT  
                      FROM ....  
                      .... ) ;
```

- Vrai si la valeur de *exp* apparaît dans le résultat de la sous requête (ou n'apparaît pas dans le cas du **NOT IN**)
- **IN** est équivalent à **=ANY**
- **NOT IN** est équivalent à **!=ANY**

```
SELECT ....  
FROM ....  
WHERE exp op ANY ( SELECT  
                      FROM ....  
                      .... ) ;
```

- Vrai s'il existe un *b* parmi les lignes renvoyées par la sous requête tel que *exp op b* soit vrai

```
SELECT ....  
FROM ....  
WHERE exp op ALL ( SELECT  
                      FROM .... ) ;
```

- Vrai si toutes les lignes *b* renvoyées par la sous requête tel que *exp op b* soit vrai

Avec $\text{op} \in \{=, <>, <, \leq, >, \geq\}$

- Sélection
- Recherche
- Ordonner les réponses

- Fonctions de groupe
- Sous-requêtes
- Opérateurs de l'algèbre relationnelle

Sous requête donnant 0,1 ou plusieurs lignes

Exemple :

Pour connaître les vols qui sont assurés par un pilote qui habite Paris.

```
SELECT      NumVol
FROM        Vol
WHERE       NumP IN (    SELECT      NumP
                        FROM        Pilote
                        WHERE      VilleP = 'Paris' );
```

- Sélection
- Recherche
- Ordonner les réponses

- Fonctions de groupe
- Sous-requêtes
- Opérateurs de l'algèbre relationnelle

Sous requête donnant 0,1 ou plusieurs lignes

Exemple :

Pour connaître le ou les pilotes qui ont le plus grand nombre d'heures de vols.

```
SELECT      NUMP
FROM        pilote
WHERE       nbhvol >= ALL (SELECT      nbhvol
                           FROM        pilote ) ;
```

ou

```
SELECT      NUMP
FROM        pilote
WHERE       nbhvol = (SELECT      max(nbhvol)
                      FROM        pilote ) ;
```

- Sélection
- Recherche
- Ordonner les réponses

- Fonctions de groupe
- Sous-requêtes
- Opérateurs de l'algèbre relationnelle

Propriétés

$\text{exp1} = \text{ANY } (\text{SELECT exp2 FROM...}) \Leftrightarrow \text{exp1 IN } (\text{SELECT exp2 FROM...})$

$\text{exp1} \neq \text{ALL } (\text{SELECT exp2 FROM...}) \Leftrightarrow \text{exp1 NOT IN } (\text{SELECT exp2 FROM...})$

$\text{exp1} < (\text{resp} \leq) \text{ ANY } (\text{SELECT exp2 FROM...})$ est vrai \Leftrightarrow
 $\text{exp1} < (\text{resp} \leq) \text{ MAX } (\text{exp2}) \text{ FROM...}$ est vrai

$\text{exp1} < (\text{resp} \leq) \text{ ALL } (\text{SELECT exp2 FROM...})$ est vrai \rightarrow
 $\text{exp1} < (\text{resp} \leq) \text{ min } (\text{exp2}) \text{ FROM...}$ est vrai

$\text{exp1} > (\text{resp} \geq) \text{ ANY } (\text{SELECT exp2 FROM...})$ est vrai \Leftrightarrow
 $\text{exp1} > (\text{resp} \geq) \text{ min } (\text{exp2}) \text{ FROM...}$ est vrai

$\text{exp1} > (\text{resp} \geq) \text{ ALL } (\text{SELECT exp2 FROM...})$ est vrai \rightarrow
 $\text{exp1} > (\text{resp} \geq) \text{ max } (\text{exp2}) \text{ FROM...}$ est vrai

- Sélection
- Recherche
- Ordonner les réponses

- Fonctions de groupe
- Sous-requêtes
- Opérateurs de l'algèbre relationnelle

Sous requêtes

- Sous requêtes Retournant au plus ligne
- Sous requêtes Multi-lignes
- Sous requêtes d'existence

- Sélection
- Recherche
- Ordonner les réponses

- Fonctions de groupe
- Sous-requêtes
- Opérateurs de l'algèbre relationnelle

Sous-requêtes d'existence

EXISTS: s'utilise dans une clause conditionnelle pour savoir s'il y a une présence ou non de lignes lors de l'utilisation d'une sous-requête.

Syntaxe:

```
SELECT ...
FROM  table1 [alias]
WHERE [NOT] EXISTS (   SELECT ...
                      FROM ... );
```

- Vrai si la sous requête retourne au moins un tuple résultat (ou ne retourne pas de tuple dans le cas du **NOT EXISTS**)
- Dans la sous requête, pour faire référence à une colonne extérieure qui a le même nom qu'une colonne intérieure, il faut la préfixer par le nom ou l'alias de la table dont elle provient.

- Sélection
- Recherche
- Ordonner les réponses

- Fonctions de groupe
- Sous-requêtes
- Opérateurs de l'algèbre relationnelle

Sous-requêtes d'existence

- Dans la sous requête, pour faire référence à une colonne extérieure qui a le même nom qu'une colonne intérieure, il faut la préfixer par le nom ou l'alias de la table dont elle provient.

```
SELECT      *
FROM        table1 aliasT
WHERE       EXISTS (   SELECT  *
                      FROM        table2 A
                      WHERE       A.idT= aliasT.idT) ;
```

- Sélection
- Recherche
- Ordonner les réponses

- Fonctions de groupe
- Sous-requêtes
- Opérateurs de l'algèbre relationnelle

Sous-requêtes d'existence

Exemple :

Pour connaître les avions qui sont conduits par au moins un pilote de Marseille.

```
SELECT      DISTINCT NumAv
FROM        Vol V
WHERE       EXISTS ( SELECT *
                      FROM Pilote P
                      WHERE P.NumP=V.NumP
                        AND P.VilleP='Marseille' );
```

S'il y'a au moins une ligne dans la table PILOTE dont villeP contient la valeur Marseille et au moins un pilote qui se trouve dans la table Pilote ($P.NumP=V.NumP$) alors la sous-requête retournera au moins un résultat. Dès lors la condition est vérifiée et la requête principale retournera les résultats de la colonne NumAv de la table Vol

- Sélection
- Recherche
- Ordonner les réponses

- Fonctions de groupe
- Sous-requêtes
- Opérateurs de l'algèbre relationnelle

Sous-requêtes d'existence

NOT EXISTS (SELECT * FROM... WHERE exp1=exp2 ...)

<-> exp1 **IS NULL OR** exp1 **NOT IN** (SELECT exp2 FROM...
WHERE exp2 IS NOT NULL...)

Exemple :

Pour connaître les avions qui n'assurent aucun vol.

```
SELECT NumAv  
FROM Avion A  
WHERE NOT EXISTS (SELECT *  
                   FROM VOL V  
                   WHERE V.NumAv=A.NumAv) ;
```

```
SELECT NumAv  
FROM Avion  
WHERE NumAv NOT IN (SELECT NumAv  
                     FROM VOL) ;
```

Ou

- Sélection
- Recherche
- Ordonner les réponses

- Fonctions de groupe
- Sous-requêtes
- Opérateurs de l'algèbre relationnelle

En résumé

- Les sous-requêtes doivent être placées entre parenthèses.
- Une sous requête ne peut avoir qu'une seule colonne dans sa clause SELECT, à moins que la requête principale ne contienne plusieurs colonnes à comparer avec celles de la sous-requête.
- La fonction ORDER BY ne peut pas figurer dans une sous-requête
- Bien réfléchir au résultat retourné par une sous-requête (une ligne, multi-lignes, multi-colonnes)